

# Acquisition of Visual Features Through Probabilistic Spike-Timing-Dependent Plasticity

Amirhossein Tavanaei:<sup>\*</sup>

*The Center for Advanced Computer Studies, University of Louisiana at Lafayette,  
Lafayette, LA 70504, USA*

Timothée Masquelier:<sup>†</sup>

*INSERM, U968, Paris, F-75012, France*

*Sorbonne Universités, UPMC Univ Paris 06, UMR-S 968, Institut de la Vision,  
Paris, F-75012, France*

*CNRS, UMR-7210, Paris, F-75012, Paris, France*

*CERCO UMR 5549, CNRS, Univ. Toulouse 3, Pavillon Baudot CHU Purpan BP 25202,  
31052 Toulouse Cedex, France*

Anthony S. Maida:<sup>‡</sup>

*The Center for Advanced Computer Studies, University of Louisiana at Lafayette,  
Lafayette, LA 70504, USA*

1

## Abstract

This paper explores modifications to a feedforward five-layer spiking convolutional network (SCN) of the ventral visual stream [Masquelier, T., Thorpe, S., Unsupervised learning of visual features through spike timing dependent plasticity. PLoS Computational Biology, 3(2), 247-257]. The original model showed that a spike-timing-dependent plasticity (STDP) learning algorithm embedded in an appropriately selected SCN could perform unsupervised feature discovery. The discovered features were interpretable and could effectively be used to perform rapid binary decisions in a classifier.

In order to study the robustness of the previous results, the present research examines the effects of modifying some of the components of the original model. For improved biological realism, we replace the original non-leaky integrate-and-fire neurons with Izhikevich-like neurons. We also replace the original STDP rule with a novel rule that has a probabilistic interpretation. The probabilistic STDP slightly but significantly improves the performance for both types of model neurons. Use of the Izhikevich-like neuron was not found to improve performance although performance was still comparable to the IF neuron. This shows that the model is robust enough to handle more biologically realistic neurons. We also conclude that the underlying reasons for stable performance in the model are preserved despite the overt changes to the explicit components of the model.

## 1 Introduction

The pattern recognition functionality found in the ventral visual pathway of the primate brain is supported by a multi-layer, spiking biological network which is able to learn to recognize new

---

<sup>\*</sup>tavanaei@louisiana.edu

<sup>†</sup>timothee.masquelier@alum.mit.edu

<sup>‡</sup>maida@cacs.louisiana.edu

<sup>1</sup>©2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

patterns. An early and highly influential model of this computation is found in [1]. They introduced a hierarchy of simple/complex cells (model neurons) that incorporated feature extraction and location invariance pooling over larger and larger receptive field sizes. The design has been used in many subsequent deep networks and is a member of the class of convolutional networks, that dates back to [2] and [3].

The model in [1] did not use spiking neurons and thus could not support spike-timing-dependent plasticity (STDP) learning, which is known to occur in many parts of the mammalian brain. Furthermore, under some conditions the recognition process in primates is so rapid that there is barely enough time for even one spike to travel from the retina to the decision region in the inferior temporal cortex [4]. The first artificial spiking neural network (SNN) that included STDP and was designed to address the speed of recognition observed in the human and primate brain is that of Masquelier and Thorpe [5]. Empirical data imposes the constraint that the majority of information used in recognition is contained in the first neural spike for any neuron in the feedforward pipeline. The paper [5] presented a model that demonstrated how this was computationally possible and the information provided by the first spike was sufficient to drive an STDP-based feature discovery algorithm to support the recognition process.

The initial work was performed on the Caltech data set which presented images in a canonical viewpoint. Later work in [6] showed that this algorithm would scale to larger data sets which showed the same object from multiple views and at multiple sizes. The recognition performance compared favorably with other approaches, specifically, [1] and [7].

One limitation of the work in [5] is that, despite the key observations underlying the design of the model, it raises new questions about biological realism that beg to be answered. Specifically, there are two limitations that the work presented here considers. First, spike generation in [5] is performed by non-leaky integrate-and-fire (IF) neurons. This may not be a serious limitation, given the short time scale under which the model operates. However, it would be prudent to examine the model’s performance using a more realistic spike-generation mechanism. This is possible by replacing the IF neuron with a simplified Izhikevich model neuron [8, 9].

The second question is why the STDP version used in [5] is effective. Put differently, can we understand why this, or alternative STDP mechanisms, perform effective feature discovery within this framework? The work in [6] shows that indeed that the STDP used in [5] is an effective feature discovery mechanism. Furthermore, a theory of how global properties of the network emerge through STDP was developed in [10].

The present paper takes preliminary steps to develop a probabilistic or Bayesian variant of the original model in [5]. In particular, we are inspired by the papers of Nessler et al. [11, 12]. Those papers show that an appropriately chosen STDP rule with a particular probabilistic interpretation, when embedded in an appropriately designed winner-take-all (WTA) network, is able to approximate online expectation maximization. Their paper also includes proof-of-concept experiments showing that implementations of the model are able to perform pattern classification.

One drawback of the model in [12] is that its trainable ‘excitatory’ connections use negative weights. This is because the trained weights converge to the log of a probability value (explained later). This gives an elegant probabilistic interpretation to the weights but since the logarithm of a probability value is negative, the weights are necessarily negative. The authors note that theoretically the weights can be shifted to positive values by choosing a positive parameter,  $c$ , equal to the magnitude of the largest negative weight in the implementation. The present paper seeks a solution where the (excitatory) weights are positive.

Although the network in [5] has five layers, it can be viewed functionally as consisting of three feedforward-connected modules: feature extraction, feature discovery, and classification. The feature extraction and discovery components are duplicated across five spatial scales. The feature discovery module uses a simple form of spike-time-dependent plasticity (STDP) based learning. We compare the IF neuron with a simplified Izhikevich regular-spiking (RS) neuron which models the spike generation dynamics of one type of neocortical pyramidal cell, the principle neuron in the neocortex.

The present research studies variants of this network when STDP is modified and when the model neurons are modified. Specifically, we modified the original implementation in [5]. The motivation with respect to STDP was to obtain some insight as to why the learning rule is able to support feature discovery and to assess the effectiveness of our new probabilistic STDP

rule in feature discovery. The motivation with regard to using different model neurons was to incrementally improve the biological realism of the model and to examine the robustness of STDP-based feature acquisition with a different model neuron. Finally, if our manipulations do not have an effect on the qualitative performance of the model, either positive or negative, then we have established that the model has some degree of robustness.

## 2 Background

STDP modifies synaptic weights in response to local timing differences between pre- and postsynaptic spikes. The existence of several variants of STDP within synapses is well established in biological networks [13]. In most cases, however, there is little consensus about its larger role in learning for biological networks. Idealized models of STDP have been studied extensively.

In recent literature, there have been at least two approaches to addressing the question of why biological STDP might be effective in learning. The first and most established approach seeks to obtain an idea of what kinds of temporal spike codes STDP produces [14, 15] and what kind of codes STDP can learn to recognize [16].

The second approach assumes that the relevant biological networks implement a Bayesian computation. Empirical studies have provided evidence that the some brain areas may perform a Bayesian analysis of sensory stimuli [17, 18]. Also, it is claimed that the spike generation process is stochastic with exponential function upon the membrane potential [19, 20]. Probabilistic interpretations of STDP could form a theoretical link to the broader area of machine learning. Nessler et al. (2009, 2013) showed that a version of the STDP rule can perform Bayesian computation in a spiking winner-take-all (WTA) network [11, 12].

Based on the Nessler et al. (2013) STDP rule, Kappel et al. (2014) developed a generic cortical microcircuit to approximately implement the hidden Markov model (HMM) learning [21]. In addition, a modified variant of this probability-based STDP has been employed in a Gaussian mixture model (GMM) training of the HMM equipped by embedded SNNs in its states [22].

The present work is inspired by the last approach and seeks to study a probabilistic rule in the context of the network used in [5].

## 3 Network Architecture

The SNN architecture in the previous study (Masquelier and Thorpe, 2007 [5]) is shown in Fig. 1. The network consists of four layers hierarchy of spiking neurons (S1-C1-S2-C2) in temporal domain. Simple cells (S) compute a linear sum (integration) while complex cells (C) compute nonlinear max pooling. The last layer (layer 5) performs a classification over visual features received from C2.

Layer S1 detects edges of the image with four orientations using intensity-to-latency conversion. The WTA between S1 cells is implemented to find the best matching orientation. C1 maps subsample S1 maps by taking the maximum response over a square neighborhood. S2 is selective to intermediate complexity visual features, defined as a combination of oriented edges. C2 takes the maximum response of S2 over all positions and scales. The maximum operation of the complex cells propagates the first spike emitted by a given group of afferent neurons because of the time-to-first-spike framework supposed in this architecture.

STDP learning occurs in the weights projecting from layer C1 to layer S2. The STDP has the effect of concentrating large synaptic weights on afferent neurons that systematically fire early, while postsynaptic spike latencies decrease. The synaptic weights are updated and duplicated at all positions and scales. The synaptic weights are considered between 0 and 1 to ensure excitatory synapses. A simplified STDP has been used in this structure in which the time window is supposed to cover the whole spike wave.

When a cell fires, local inhibition prevents other cells in that layer from firing at the same scale and within a square neighborhood of the firing position.

The final membrane potential measured in the C2 cells (and is contrast invariant) is used for training the radial basis function (RBF) classifier. Images are processed by the network one by

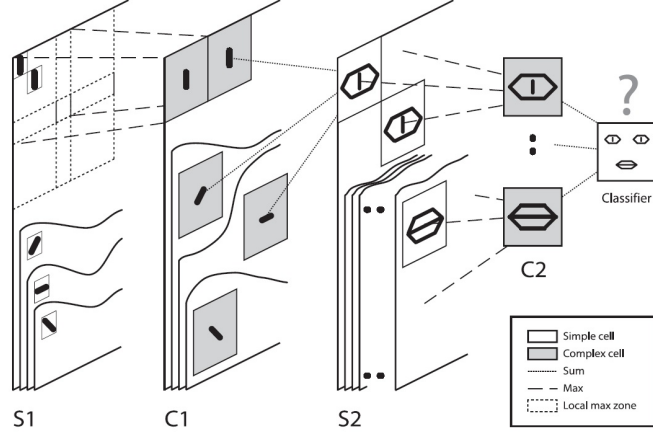


Figure 1: 5-layer hierarchical spiking neural network developed in [5]. There is one S1-C1-S2 pathway for each processing scale. C2 cells select the maximum response received from S2. Finally, visual features are classified by the last layer. STDP learning is applied to the synapses connecting C1 and S2 neuron layers. The synaptic weight change in the SNN proposed in [5] is computed by a factor of  $w(1 - w)$ . In the current investigation, we change the STDP rule and neuron models to study the effect on performance. (from [5])

one and the neurons in each layer are allowed to fire only once. Finally, the classifier layer utilizes RBF classification to recognize the patterns according to the visual features received from C2.

## 4 The STDP rules tested

In a later section, we will examine network classification performance using a novel STDP rule. Here, we explain the original STDP rule used in [5], along with the new rule. Neither rule needs to compute the exact time difference between pre- and postsynaptic spikes.

### 4.1 The original STDP rule

The original STDP rule given in [5] is shown below.  $w_{ki}$  denotes the weight from presynaptic neuron  $i$  to postsynaptic neuron  $k$ .

$$\Delta w_{ki} = \begin{cases} a^+ \cdot w_{ki} \cdot (1 - w_{ki}), & t_k^f - t_i^f \geq 0 \\ -a^- \cdot w_{ki} \cdot (1 - w_{ki}), & \text{otherwise} \end{cases} \quad (1)$$

The first case describes LTP and the second describes LTD.  $t_i^f$  and  $t_k^f$  denote firing times of units  $i$  and  $k$ , respectively. Weights fall in the range  $(0, 1)$ . Note that the quantity  $w_{ki} \cdot (1 - w_{ki})$  is the slope of the sigmoidal function. Therefore, if  $a^+ = 1$  and  $a^- = 1$ , weight changes will be largest when  $w_{ki}$  is 0.5 and weight changes will become arbitrarily small as  $w_{ki}$  approaches the bounds 0 or 1. In the most of simulations, the magnitudes of amplification parameters  $a^+$  and  $a^-$  are kept in a ratio of 4/3.

### 4.2 Probabilistic STDP variant

The proposed new rule yields weights with a probabilistic interpretation after convergence. Our new rule is modified from [12]. The rule from [12], when embedded in an appropriate winner-take-all network with Poisson spiking neurons can approximate expectation maximization learning. It is given below.

$$\Delta w_{ki} = \begin{cases} e^{-w_{ki}} - 1, & 0 < t_k^f - t_i^f < \epsilon \\ -1, & \text{otherwise} \end{cases} \quad (2)$$

LTP occurs if the presynaptic neuron fires briefly (e.g., within  $\epsilon = 10$  ms) before the postsynaptic neuron. Otherwise LTD occurs. The rule is unusual in that the magnitude of the LTP weight adjustment is exponentially related to the current value of the weight. Additionally, weight values are always negative. After convergence, the weight value is equal to the log of the probability that the presynaptic neuron will have fired within the  $\epsilon$  interval, given that the postsynaptic neuron has fired.

Our SNN uses positive weights for the excitatory neurons in the feedforward layers and therefore the rule (2) is not applicable. Therefore, we modified (2) to obtain (3) below.

$$\Delta w_{ki} = \begin{cases} a^+ \cdot e^{-w_{ki}}, & t_k^f - t_i^f \geq 0, \quad (\text{LTP}) \\ -a^-, & \text{otherwise,} \quad (\text{LTD}) \end{cases} \quad (3)$$

where  $a^+$  and  $a^-$  are the amplification parameters used in (1). The synaptic weights are constrained to be positive during the training process.

### 4.3 Probabilistic perspective

To interpret the behavior of the STDP rule probabilistically, we follow the approach of Nessler et al. (2013), p. 21, formula 28 [12]. At equilibrium, the expected weight change will be zero. This can be written and then simplified as shown below. The notation  $z_k = 1$  means that the postsynaptic neuron fired on a given stimulus presentation. The notation  $y_k = 1$  means that the presynaptic neuron fired before the postsynaptic neuron on the same stimulus presentation.

$$\begin{aligned} E[\Delta w_{ki}] &= 0 \Leftrightarrow \\ a^+ \cdot p_{\mathbf{w}}^*(y_i = 1|z_k = 1)e^{-w_{ki}} - a^- \cdot p_{\mathbf{w}}^*(y_i = 0|z_k = 1) &= 0 \Leftrightarrow \\ a^+ \cdot p_{\mathbf{w}}^*(y_i = 1|z_k = 1)e^{-w_{ki}} + a^- \cdot p_{\mathbf{w}}^*(y_i = 1|z_k = 1) &= a^- \Leftrightarrow \\ e^{-w_{ki}} &= \frac{a^- - a^- \cdot p_{\mathbf{w}}^*(y_i = 1|z_k = 1)}{a^+ \cdot p_{\mathbf{w}}^*(y_i = 1|z_k = 1)} \Leftrightarrow \\ w_{ki} &= \ln\left(\frac{a^+}{a^-} \cdot \frac{p_{\mathbf{w}}^*(y_i = 1|z_k = 1)}{1 - p_{\mathbf{w}}^*(y_i = 1|z_k = 1)}\right) \Leftrightarrow \\ w_{ki} &= \ln\left(\frac{a^+}{a^-}\right) + \ln(\text{Odds Ratio}) \end{aligned} \quad (4)$$

where,  $p^*$  denotes an equilibrium probability.

The synaptic weight,  $w_{ki}$ , in (4) specifies the log odds (logit) of the probability. The probability,  $p^*$ , denotes the casual effect of presynaptic neuron in postsynaptic spike in which the STDP rule is triggered. The odds ratio is the ratio of the probability that the event of interest occurs to the probability that it does not [23]. The event of interest is the presynaptic activation right before the postsynaptic neuron fires. Positive log odds ( $p_{\mathbf{w}}^*(y_i = 1|z_k = 1) \geq \frac{a^-}{a^- + a^+}$ ) specifies a probable condition in which the synaptic weight is subject to LTP. The parameter  $\frac{a^-}{a^- + a^+}$  determines a stochastic threshold for postsynaptic firing. The negative log odds triggers LTD until the synaptic weight reaches its minimum efficacy. The minimum efficacy is when the presynaptic neuron does not have an effect on the postsynaptic membrane potential (disconnected synapse).

A number of previous empirical studies have provided evidence that Bayesian analysis of sensory stimuli occurs in the brain [17, 18, 24]. In Bayesian inference, the hidden causes are inferred using prior knowledge and the likelihood of the observations to obtain a posterior probability. In the case of two classes  $M_1$  and  $M_2$ , the posterior probability of class  $M_1$  is introduced by the sigmoid function as follows [25]

$$p(M_1|x) = \frac{1}{1 + e^{-a}} \quad (5)$$

where  $a$  is

$$a = \ln \frac{p(x|M_1)p(M_1)}{p(x|M_2)p(M_2)} \quad (6)$$

which is interpreted as log odds by

$$a = \ln \frac{p(M_1|x)}{p(M_2|x)} = \ln \frac{p(M_1|x)}{1 - p(M_1|x)} \quad (7)$$

Equation (7) is analogous to the synaptic weight in (4).

#### 4.4 Limitations and convergence

We consider the maximum possible weight value after a series of the LTP events during training. Does an excitatory synaptic weight converge to some constant or grow without bound as a function of LTP events? To address this, we start with initial weight,  $w^0$ . The next synaptic weights after performing  $n$  LTP rules are  $w^1, w^2, \dots, w^n$ . According to (3),

$$w^{i+1} = w^i + a^+ \cdot e^{-w^i} \quad (8)$$

The maximum weight change happens when  $w^i = 0$ . So, if we suppose  $w^0 = 0$  (maximum change), the final synaptic weight is obtained by:

$$w^n = w^0 + a^+ \cdot e^{-w^0} + \dots + a^+ \cdot e^{-w^{n-1}} = a^+ \cdot \sum_{i=0}^{n-1} e^{-w^i} \quad (9)$$

And to find an upper bound for the synaptic weight in  $n$  LTP operations, we have:

$$\begin{aligned} w^1 &= a^+, \\ w^2 &= a^+ + a^+ \cdot e^{-a^+} < 2a^+, \\ w^3 &< 2a^+ + a^+ \cdot e^{-2a^+} < 3a^+, \\ \text{and...} \\ w^{n-1} &< (n-2)a^+ + a^+ \cdot e^{-(n-2)a^+} < (n-1)a^+ \end{aligned} \quad (10)$$

Therefore,

$$w^n < a^+ \cdot \sum_{i=0}^{n-1} e^{-ia^+} = a^+ \cdot \frac{1 - e^{-na^+}}{1 - e^{-a^+}} \quad (11)$$

Finally, a synaptic weight after infinite iterations of the LTP adjustments (with  $0 < a^+ < 1$ ) will fall in range  $(0, 1.582)$  where, 1.582 is an upper bound if  $a^+ = 1$ .

### 5 Izhikevich's Neuron Model

Izhikevich's model for the RS spiking neuron concisely, but accurately, captures the spike-generation dynamics of one subtype of pyramidal neuron [8, 9]. We used a simplified version of this so that it would be compatible with the SNN implementation (We only use the first generated spike). Its spike-generation dynamics is specified in Equation Set (12) using state variables  $V$  and  $U$ .

$$C \frac{dV}{dt} = k(V - V_{\text{rest}})(V - V_{\text{th}}) - U + I_{\text{tot}} \quad (12a)$$

$$\frac{dU}{dt} = a[b(V - V_{\text{rest}}) - U] \quad (12b)$$

$V$  denotes the membrane potential and  $U$  specifies a recovery factor and keeps the membrane potential near the resting value,  $V_{\text{rest}}$ .

The many different parameters: capacity ( $C$ ), threshold ( $V_{\text{th}}$ ),  $a$ ,  $b$ ,  $c$ ,  $d$ , and  $k$  customize the spike-generation dynamics. The model parameters to obtain an RS spiking neuron are:  $a = 0.03$ ,  $b = -2$ ,  $U_0 = 0$ ,  $V_{\text{rest}} = 0$ ,  $V_{\text{th}} = 0.49$ ,  $C = 100$ ,  $K = 0.7$ ; where  $V_{\text{th}}$  and  $V_{\text{rest}}$  were changed according to the model requirements.

The spike time occurs when  $V$  reaches  $V_{\text{th}}$ .  $I_{\text{tot}}$  is set to the net synaptic input.

### 6 Experiments and Results

We performed two experiments. Both experiments compared four network architectures. Specifically, we studied four types of the 5-layer SNNs as follows:

- **SNN 1:** The original 5-layer convolutional network of IF spiking neurons in [5] with the original parameters.

- **SNN 2:** The SNN equipped with our probabilistic STDP learning rule.
- **SNN 3:** The original SNN using Izhikevich-like neurons instead of IF neurons.
- **SNN 4:** The SNN using Izhikevich-like neurons and equipped with the probabilistic STDP rule.

In all of the SNNs, the learning parameter  $a^+ = 2^{-6}$  at the beginning and is multiplied by 2 every 400 postsynaptic spikes, until a maximum value of  $2^{-2}$ .  $a^-$  is adjusted such that  $a^+/a^- = 4/3$ .



Figure 2: Sample images from the Caltech dataset. The top row shows face examples and the bottom row shows background examples.

## 6.1 Experiment 1

### 6.1.1 Method

To compare with the original work [5], we evaluated the SNN using a subset of the Caltech dataset (<http://www.vision.caltech.edu/archive.html>) containing faces (positive class) and background images (negative class) that was used in the original model. We also compared motorbikes to background images. Images were converted to gray-scale as described in [5]. Examples images for face and background appear in Fig. 2 (more examples, including motorbikes, are given in [5]). The dataset was equally divided into training and testing sets.

In one simulation, the SNNs were applied on the face/background (non-face) training samples. In a second simulation, the bike/background data was used.

These simulations produced ten class-specific C2 features. Each C2 cell represents a combination of edges in respect to an input image. A representation of the input images can be reconstructed by convolving the weight matrix with a set of kernels representing oriented bars [5]. During the training process, initial iterations do not show a clear reconstructed preferred stimulus. After training, the C2 features are able to represent the face images. Thus, the C2 features developed selectivity to face features. Learning was stopped after 1000 iterations while the weight matrices were saved after each 100 iterations to track the learning process.

The trained network was evaluated using the testing set. Following [5], we used two measures for evaluating the model: 1) the accuracy rate when the false positive rate equals the missed rate (equilibrium point); and, 2) A term of area under the receiver operator characteristic (ROC).

### 6.1.2 Results and discussion

Fig. 3 shows the maximum accuracy rate obtained from the SNN models (**SNN 1** through **SNN 4**). It also includes standard errors of the mean. These were computed by running each simulation nine times ( $n = 9$ ) with a different set of initial values for the weights. Different subsamples for training and testing data were not used in this first experiment but they were used in second

experiment. Fig. 3 shows a trend that the probabilistic STDP rule gives better accuracy regardless of whether the IF neuron (**SNN1** vs **SNN2**) or the Izhikevich-like neuron is used (**SNN3** vs **SNN4**). The overall performance for the motorbikes data was lower. Fig. 4 illustrates the same pattern of results for the maximum ROC obtained from the SNN models for both face and motorbike datasets.

Two-tailed t-tests were performed to assess the statistical significance of the accuracy trends. The following comparisons were performed: **SNN1** versus **SNN2**, **SNN3** versus **SNN4**, and **SNN1** versus **SNN4** for both faces and motorbikes. The results are shown in Figs. 3 and 4 in terms of  $p < 0.1$  (\*),  $p < 0.05$  (\*\*),  $p < 0.01$  (\*\*\*), and  $p < 0.005$  (\*\*\*\*). The differences reached more significance for the network that used Izhikevich-like neurons. Thus, we claim that our probabilistic variant STDP rule significantly improves performance when Izhikevich-like neurons are used, either with faces or motorbikes.

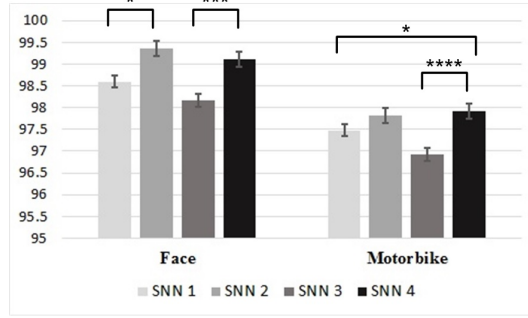


Figure 3: Summary of performance for face and bike recognition in term of accuracy rate in equilibrium point (%).

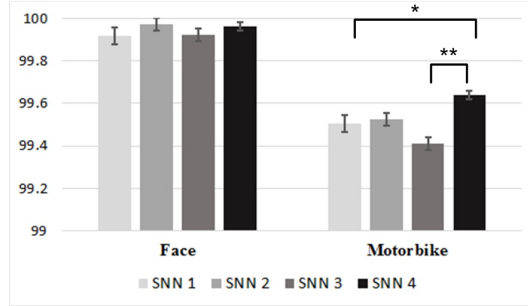


Figure 4: Summary of performance for face and bike recognition in term of area under the ROC curve (%).

Fig. 5 shows reconstructions for the ten C2 cells on the face data for the four SNNs. They indicate that face-like features were discovered by all SNN variants. Thus, that property of the original SNN in [5] is shown to be robust under the changes introduced in this experiment. Fig. 6 shows the reconstructions for a selected face feature during the training process (100 to 1000 iterations) for each of the SNNs. Qualitatively, the results are similar for all of the SNNs.

Fig. 7 shows the accuracy results for all of SNNs applied to the faces data. The most salient trend is that **SNN 3** and **SNN 4**, which both use the Izhikevich-like units, perform better relatively early in the training, at iterations 200 and 300, but the advantage is lost later in the training. Also, **SNN 3** seems to perform slightly worse than the others. Standard errors were not computed, but this was done in Experiment 2.



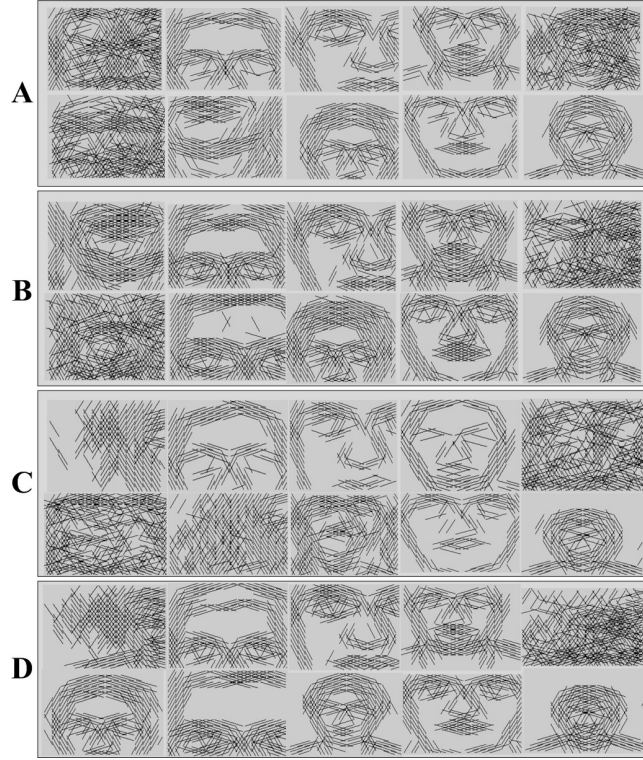


Figure 5: Reconstructions for the 10 features having developed selectivity to face features. (A) through (D) specify the reconstructions of **SNN 1** through **SNN 4** respectively.

## 6.2 Experiment 2

The results in Experiment 1 need further exploration because, first, the results are possibly compromised by ceiling effects (especially for the faces data) and, second, we did not perform cross-validation by using different samples of training and testing data.

### 6.2.1 Method

To address the possible ceiling effect, we used noisy test images (Gaussian noise with zero mean and standard deviation of 0.3) for the testing phase. The purpose was to reduce the accuracy rate so that ceiling effects were no longer an issue. Five sets of runs ( $n = 5$ ) were performed for both faces and motorbikes. Both testing and training images sets had 218 items. For each simulation, 175 items were sampled from each set to be used in that simulation. Standard errors were computed for each of 40 data points.

### 6.2.2 Results and discussion

Fig. 8 shows the face results for all of SNNs (**SNN 1** through **SNN 4**) including standard errors. Fig. 9 shows the results for motorbikes. In both figures the accuracy rates are below 97 percent, so using the noisy images eliminated the complication of a ceiling effect. For the faces (Fig. 8), all of the trends shown in Fig. 7 are preserved in these results. Most notably, the trend where the probabilistic STDP variant improves accuracy regardless of whether IF units or Izhikevich-like units are used still persists. For the motorbikes, the proposed probabilistic STDP performs better; however, the accuracy performances are close across all manipulations. From the data taken as a whole, we can also conclude that the original model is robust across the manipulations.

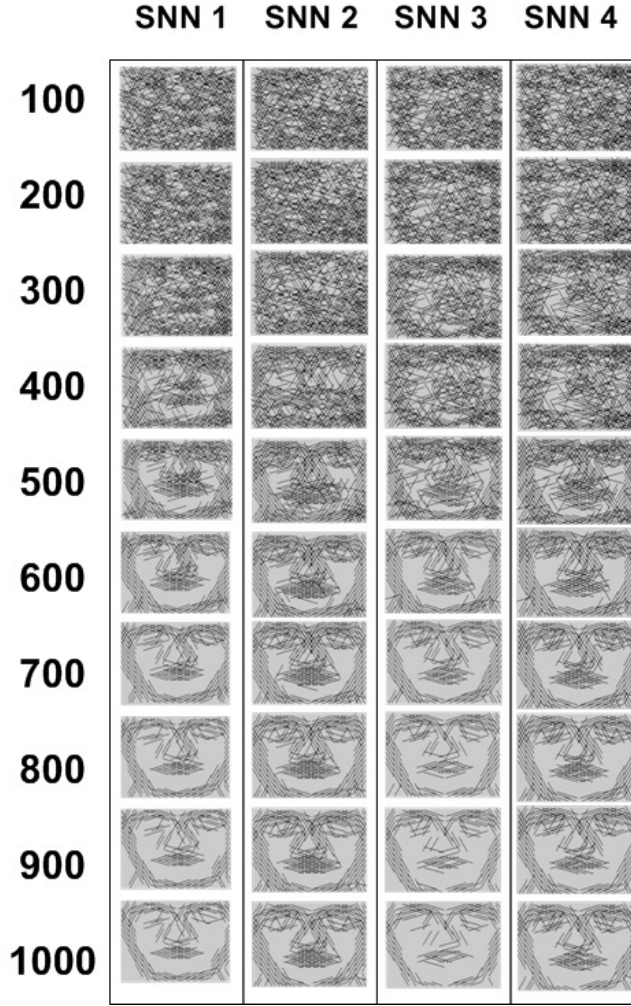


Figure 6: Reconstructions of a selected face feature for **SNN 1** through **SNN 4** during the training process in 100 through 1000 iterations.

### 6.3 Control Experiments

Experiments 1 and 2 strongly suggest that the proposed probabilistic STDP rule improves the network performance in comparison to the original one. We ran two additional control experiments to improve our confidence in this result. The first experiment controlled for weight selection and the second experiment tested several other object categories.

Figs. 5 and 6 show more selected weights for probabilistic STDP (lines in face features). Hence, is the advantage of the probabilistic rule because it selected more weights? To address this, ten experiments with different  $a^+/a^-$  ratios were run on the face recognition task to assess the effect of the number of selected weights on accuracy. Fig. 10 shows the accuracy rate versus average number of the selected weights. This figure shows that the probabilistic rule consistently outperforms the classic rule for a given number of selected weights.

Additionally, a subset of the 3D-Object images provided by Savarese et al. at CVGLab, Stanford University [26] was used for new experiments. The accuracy rates of **SNN 1** through **SNN 4** are shown in Fig. 11. This figure shows better performance of probabilistic STDP (**SNN 2** and **SNN 4**) than original one (**SNN 1** and **SNN 3** respectively) in 12 out of 16 cases.

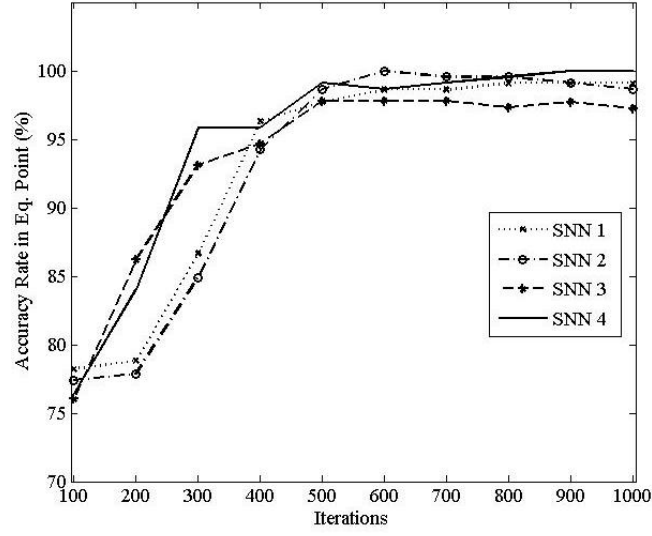


Figure 7: Accuracy rate of **SNN 1** through **SNN 4** at equilibrium point in 100 through 1000 learning iterations. (Experiment 1)

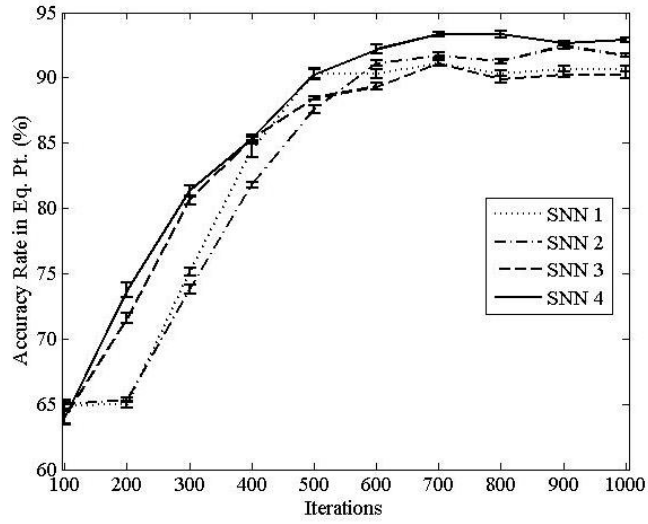


Figure 8: Accuracy of face/non-face recognition for noisy images applied to **SNN 1** through **SNN 4** for 100 through 1000 learning iterations. (Experiment 2)

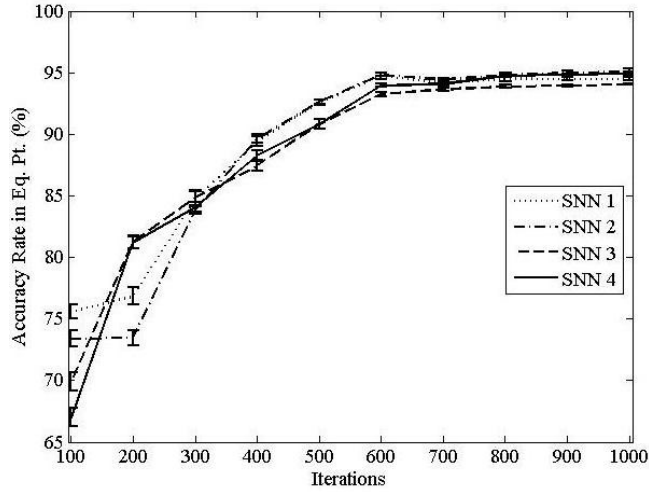


Figure 9: Accuracy rates of motorbike/non-motorbike recognition for noisy images applied to SNN 1 through SNN 4 for 100 through 1000 learning iterations. (Experiment 2)

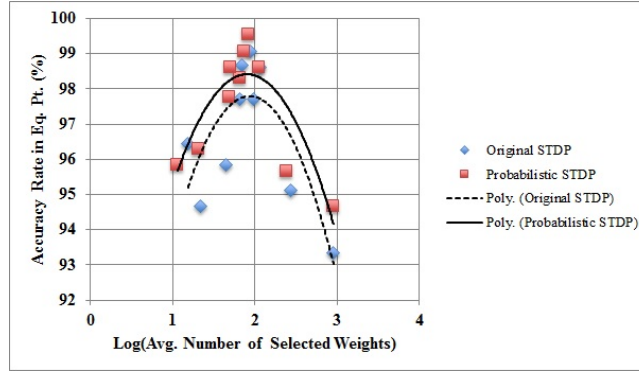


Figure 10: Accuracy rates of SNN 1 and SNN 2 versus average number of selected weights after training. Data points were obtained by 10 different training runs for each SNN with  $a^+/a^- = 4/9, 4/8, 4/7, 4/6, 4/5, 4/4, 4/3, 4/2, 4$ , and 8. The trend lines show better performance of the probabilistic variation of the STDP introduced in this paper.

## 7 Conclusion

This paper studied the performance of the spiking network given in [5] when the original STDP learning rule was replaced with a novel rule that had a probabilistic interpretation and also when the model neuron was changed from an IF neuron to an Izhikevich-like neuron.

Our findings are the following. First, our experiments introduced a probabilistic STDP variant which improved the accuracy rate. Most notably, the magnitude of the weight adjustments for LTP was an exponential function of the weight magnitude. This is an entirely different type of rule. Second, when we replaced the non-leaky IF neurons with Izhikevich-like neurons, accuracy performance remained robust. This is significant because the IF neuron is the simplest possible spike generator whereas the Izhikevich neuron has a much more complicated spike generation mechanism.

Our main conclusion is that our probabilistic variant of STDP improved performance for instances of the model that used either IF neurons or Izhikevich-like neurons. A secondary conclusion is that the original model is robust against more biologically realistic neurons (Izhikevich-like RS neurons). Although Izhikevich-like neuron did not improve the performance, considering the nature of the manipulations, new knowledge has in fact been obtained. We are not aware of other work that has studied variations of this model.

A priori, one would expect that the most likely effect of any drastic change, like those listed above, would impair network performance, but instead, the performance remained robust. Future researchers experimenting with this model, for instance to build a deep spiking network, can have some measure of confidence that the performance of the individual S/C modules will remain robust when placed in a larger context.

## References

- [1] Serre, T., Wolf, W., Bileschi, S., Riesenhuber, M., Poggio, T., Robust object recognition with cortex-like mechanisms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3), 411-426, 2007.
- [2] LeCun, Y., Bengio, Y., Convolutional networks for images, speech, and time series. In *Handbook of brain theory and neural networks (1st ed)*, 255-258, MIT Press, 1995.
- [3] Fukushima, K., Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36, 193-202, 1980.
- [4] Thorpe, S., Fize, D., Marlot, C., Speed of processing in the human visual system. *Nature*, 381, 520-522, 1996.
- [5] Masquelier, T., Thorpe, S. J., Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Computational Biology*, 3(2), e31, 247-257, 2007.
- [6] Kheradpisheh, S. R., Ganjtabesh, M., Masquelier, T., Bio-inspired Unsupervised Learning of Visual Features Leads to Robust Invariant Object Recognition. arXiv preprint arXiv:1504.03871, 2015, to appear in *Neurocomputing*.
- [7] Krizhevsky, A., Sutskever, I., Hinton, G., ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems (NIPS) 25*, 1097-1105, 2012.
- [8] Izhikevich, Eugene M., Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6), 1569-1572, 2003.
- [9] Izhikevich, Eugene M., *Dynamical systems in neuroscience*. MIT press, 2007.
- [10] Masquelier, Timothée, and Simon J. Thorpe, Learning to recognize objects using waves of spikes and spike-timing-dependent plasticity. *The 2010 International IEEE Joint Conference on Neural Networks (IJCNN)*, 2010.
- [11] Nessler, B., Pfeiffer, M., Maass, W., STDP enables spiking neurons to detect hidden causes of their inputs. *Advances in neural information processing systems*, 1357-1365, 2009.
- [12] Nessler, B., Pfeiffer, M., Buesing L., Maass, W., Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS Computational Biology*, 9(4): e1003037, 1-30, 2013.
- [13] Markram, H., Gerstner, W., Sjöström, P. J., Spike-timing-dependent plasticity: A comprehensive overview. *Frontiers in Synaptic Neuroscience*, 4(2), 1-3, 2011.
- [14] Song, Sen, and Larry F. Abbott, Cortical development and remapping through spike timing-dependent plasticity. *Neuron*, 32(2), 339-350, 2001.
- [15] Guyonneau, R., VanRullen, R., Thorpe, S., Neurons tune to the earliest spikes through STDP. *Neural Computation*, 17, 858-879, 2005.
- [16] Masquelier, Timothée, Rudy Guyonneau, and Simon J. Thorpe, Competitive STDP-based spike pattern learning. *Neural computation*, 21(5), 1259-1276, 2009.
- [17] Doya Kenji, *Bayesian brain: Probabilistic approaches to neural coding*, MIT press, 2007.

- [18] Körding, Konrad P., and Daniel M. Wolpert, Bayesian integration in sensorimotor learning. *Nature*, 427, 244-247, 2004.
- [19] Rezende, D. J., Wierstra, D., Gerstner, W., Variational learning for recurrent spiking networks. *Advances in Neural Information Processing Systems*, 136-144, 2011.
- [20] Jolivet, R., Rauch, A., Lscher, H. R., Gerstner, W, Predicting spike timing of neocortical pyramidal neurons by simple threshold models. *Journal of computational neuroscience*, 21(1), 35-49, 2006.
- [21] Kappel, David, Bernhard Nessler, and Wolfgang Maass, STDP installs in winner-take-all circuits an online approximation to hidden Markov model learning. *PLoS Computational Biology*, 10(3): e1003511, 1-22, 2014.
- [22] Tavanaei, Amirhossein, and Anthony S. Maida, Studying the interaction of a hidden Markov model with a Bayesian spiking neural network. *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, Boston, MA, USA, 2015.
- [23] Bland, J. Martin, and Douglas G. Altman, Statistics Notes: The odds ratio. *BMJ*, 320(7247), 1468, 2000.
- [24] Rao RPN, Olshausen BA, Lewicki MS, *Probabilistic Models of the Brain*. MIT Press, 2002.
- [25] Bishop, Christopher M., *Pattern recognition and machine learning*. Springer, New York, 2006.
- [26] Savarese S., Fei-Fei L., 3D generic object categorization, localization and pose estimation, *IEEE 11th International Conference on Computer Vision (ICCV)*, 18, 2007.

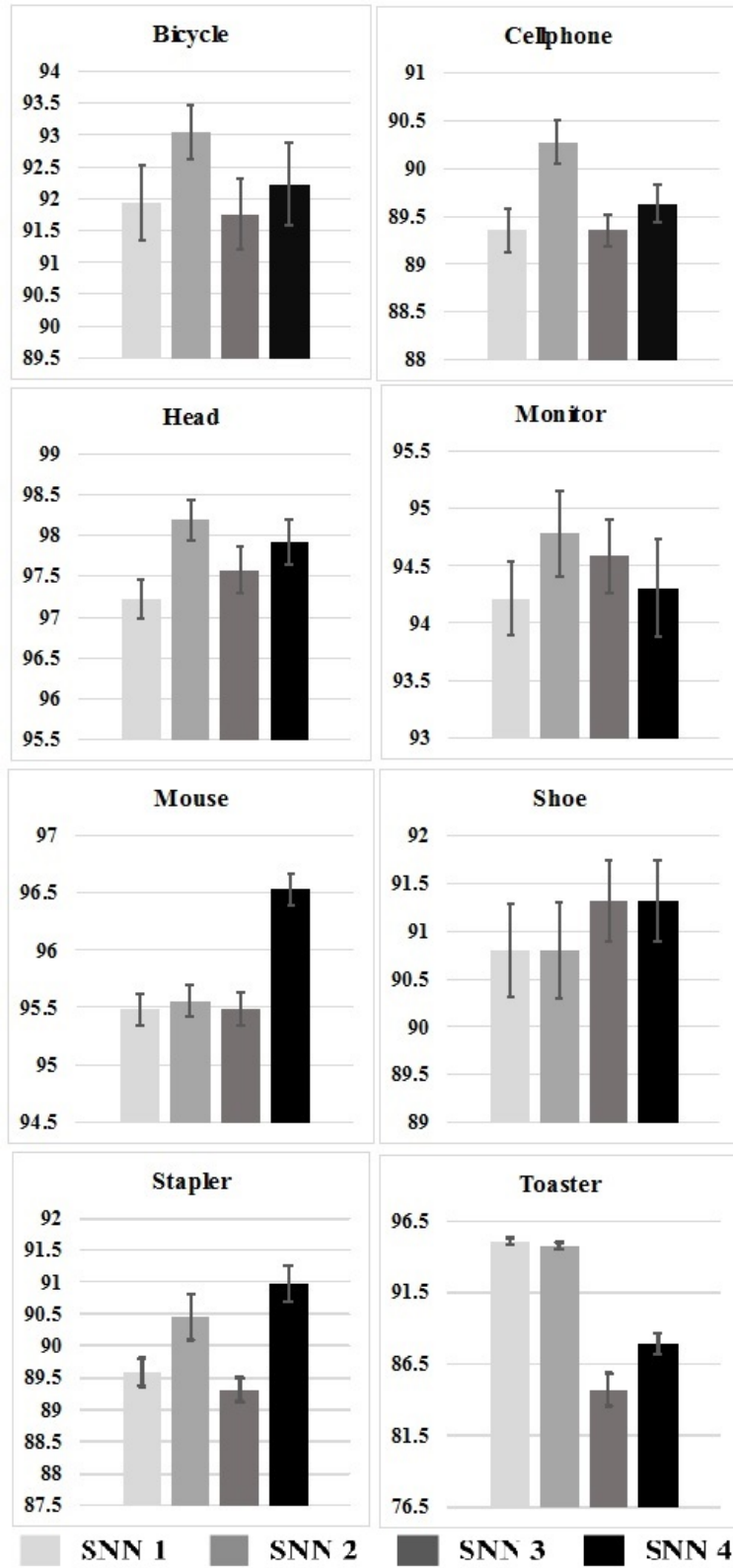


Figure 11: Accuracy rates of the SNNs for recognizing eight sets of 3D-objects provided in [26].